Mathias Anneken*, Manjunatha Veerappa*, Marco F. Huber, Christian Kühnert, Felix Kronenwett and Georg Maier

Explainable AI for sensor-based sorting systems

https://doi.org/10.1515/teme-2022-0097
Received October 31, 2022; accepted January 27, 2023
published online February 17, 2023

Abstract: Explainable artificial intelligence (XAI) can make machine learning based systems more transparent. This additional transparency can enable the use of machine learning in many different domains. In our work, we show how XAI methods can be applied to an autoencoder for anomaly detection in a sensor-based sorting system. The setup of the sorting system consists of a vibrating feeder, a conveyor belt, a line-scan camera and an array of fast-switching pneumatic valves. It allows the separation of a material stream into two fractions, realizing a binary sorting task. The autoencoder tries to mimic the normal behavior of the nozzle array and thus can detect abnormal behavior. The XAI methods are used to explain the output of the autoencoder. As XAI methods global and local approaches are used, which means we receive explanations for both a single result and the whole autoencoder. Initial results for both approaches are shown, together with possible interpretations of these results.

Keywords: autoencoder; explainable artificial intelligence; sensor-based sorting.

1 Introduction

Machine learning and artificial intelligence (AI) are solving problems and challenges from different domains in high quality. In particular, deep learning, such as neural networks, have been shown to achieve state-of-the-art performance on a wide range of applications [1–3]. However, these models are considered black-boxes because of their complex internal structure of the network. That means, a user is not necessarily able to understand how the models generate the results. In order to strengthen the trust in the models, transparency is needed. This will improve the acceptance and thus enable the usage in productive environments. Explainable artificial intelligence (XAI) is one possibility to give insights into complex models, thus supporting developers and end-users in their actual work.

An increasingly important prerequisite for real-world applications supported by AI is the ability to describe complex mathematical functions in predictive models in a way that is understandable to humans [4], e.g., in order to fulfill regulatory obligations. This promotes trust and acceptance of AI processes. One such real-world AI application that is of interest to us is sensor-based sorting. Sensor-based sorting is an established technology in many industrial fields, with growing importance for the physical sorting of materials using sensor technology. Besides its application for sorting of foodstuffs and agricultural products, industrial minerals, and quality assurance in production processes, it is considered a key technology for achieving a circular economy [5]. Hence, quality control of sensor-based sorting systems themselves is necessary to ensure the production of high-quality material streams and also to prevent systems from damage and defects.

The necessity of quality control in sensor-based sorters has led to the development of AI models for the task of anomaly detection. However, the major drawback for the acceptance of such anomaly detection models is the lack of transparency. The models are able to detect anomalies in the system, but fail to provide the reason behind their decisions. Hence, to make the decisions of such models more interpretable, we employ XAI methods. That means, in this work, we show how XAI methods increase the transparency of AI systems being utilized to detect anomalies in sensor-based sorting systems and give insights into the reasoning behind the AI decisions.

The structure of the paper is as follows: In Section 2, state-of-the-art XAI methods on time-series and unsupervised models are reviewed. The foundation towards sensor-based sorting systems, anomaly detection, selected XAI methods, and the pipeline for generating XAI explanations on an unsupervised model are described in Section 3. Section 4 provides an experimental setup used for sorting

***Corresponding authors: Mathias Anneken and Manjunatha Veerappa**, Fraunhofer IOSB, Karlsruhe, Germany; and Fraunhofer Center for Machine Learning, E-mail: mathias.anneken@iosb.fraunhofer.de (M. Anneken), manjunatha.veerappa@iosb.fraunhofer.de (M. Veerappa). https://orcid.org/0000-0002-8434-0082 (M. Anneken). https://orcid.org/0000-0002-5824-2905 (M. Veerappa)
Marco F. Huber, Department Cyber Cognitive Intelligence (CCI), Fraunhofer IPA, Stuttgart, Germany; and Institute of Industrial Manufacturing and Management IFF, University of Stuttgart, Stuttgart, Germany. https://orcid.org/0000-0002-8250-2092
Christian Kühnert, **Felix Kronenwett and Georg Maier**, Fraunhofer IOSB, Karlsruhe, Germany; and Fraunhofer Center for Machine Learning, Karlsruhe, Germany

systems, anomaly detection, and explanation after which the initial results of the experiment are illustrated. Finally, we conclude in Section 5 with a short outlook for possible future work.

2 State of the art – explainable artificial intelligence

In this section, the relevance of XAI in general is highlighted. Furthermore, the applicability of XAI to time-series data as well as unsupervised models is described.

In recent years, machine learning, in particular deep learning, has become increasingly successful in sectors such as healthcare, automotive, production, etc. [6]. However, the resulting systems are often opaque and thus, considered black-box models [1–3]. Hence, in order to be used and accepted in an actual productive environment, there is the need to build trust into the models. Therefore, it is vital to understand why a model makes certain predictions. This can be accomplished either with the help of XAI or creating models that are inherently interpretable in the first place. XAI models provide a balance between performance and interpretability over inherently interpretable models, allowing us to use advanced AI techniques while still maintaining a level of transparency and accountability [7]. Hence, this study focuses on the former case.

2.1 XAI on time-series

Since most of the data in the production domain are obtained from sensors periodically, they can be considered as time-series. This implies that the applicability of XAI on time-series data is necessary. The majority of XAI approaches are very focused on tabular, images, and text data [8]. However, Schlegel et al. presented in [8] that XAI approaches, like local interpretable model-agnostic explanation (LIME) and Shapley additive explanations (SHAP) (see Section 3.3), also work well on univariate time-series. Similar to Schlegel et al. [8], Veerappa et al. also illustrated in [4] that XAI approaches, which were initially developed for images, text, and tabular data, can be used for multivariate time-series. Similarly, Burkart et al. demonstrated in [9] how existing approaches can be applied to multivariate time-series data. Their main objective is to make the supervised black-box model transparent and understandable to the user.

Another study of XAI on time-series [10] describes a methodology to apply interpretable association rule mining approaches on time-series data. The authors explore and evaluate different interpretable rule mining approaches to mimic the behavior of the supervised classification model.

Further, a thorough overview of the various methods applied in the field of explainable supervised machine learning is provided in a survey by Burkart and Huber [11]. On the one hand, they present a structured overview of XAI approaches to supervised classification problems. On the other hand, they showcase the need for an assessment of explainability.

In the production domain, Sofianidis et al. [12] presented an overview of XAI techniques to increase the transparency of AI models. They classified different XAI techniques based on the explanation generating mechanism, the type of explanation, the scope of explanation, etc., as well as different metrics based on fidelity, unambiguity, etc., for their evaluation. Overall, they come to the conclusion that the integration of XAI into the production domain or manufacturing processes will be paramount for the transition into the fifth industrial revolution.

2.2 XAI on unsupervised models

The majority of the studies listed above is highly focused on supervised models. Supervised models [13] are trained with a dataset consisting of labelled data points, being a combination of a set of features and an associated target label. In our work, we employ an unsupervised model (model trained without target labels) for anomaly detection in a sorting system. An autoencoder, a type of an artificial neural network used to learn efficient encodings in an unsupervised learning process, is utilized for this task. This implies that the relevance of XAI on unsupervised models—specifically an autoencoder—is essential. For this purpose, Friedman Antwarg et al. proposes a methodology in [14], where XAI methods such as KernelSHAP can be used to make an unsupervised black-box model transparent. This methodology explains the anomaly detected by an autoencoder by producing one explanation per single instance, making this approach local. In contrast, Gnoss and Tropmann-Frick [15] propose a global approach that can be applied to unsupervised models as well. This approach produces both a local explanation as well as a global one.

According to another study [16], the decisions of an unsupervised model such as autoencoder for computer anomaly detection can be explained and further improved using an existing XAI method like KernelSHAP. The authors demonstrated that with the help of XAI, they were able to explain or interpret the results of an autoencoder model while simultaneously improving the overall performance of the model.

3 Foundations

In this section, the used sensor-based sorting system, autoencoder for anomaly detection, and XAI methods are introduced. First, the functional principle of sensor-based sorting is illustrated. Afterwards, a brief description of the components of autoencoders is given. Next, the XAI terminology is explained. Lastly, the adapted XAI methods for our application are described in detail.

3.1 Sensor-based sorting

State-of-the-art sensor-based sorting systems are used to classify individual particles in a material stream and physically separate the material feed into predefined classes. A common task is to remove low-quality entities from a material stream. Possible applications include quality control in production processes, the agricultural and food sector, minerals processing, and pre-treatment processes for recycling, such as sorting of plastics and construction and demolition waste.

A schematic overview of a sensor-based sorting system is shown in Figure 1. Bulk material particles are transported on a conveyor belt, preferably normally distributed along the belt width. After discharge from the belt, the particles pass the inspection line and are detected by one or multiple sensors. The sensors are selected on the basis of the specific sorting task. Usually, the systems utilize line-scanning sensors. The acquired image data is processed with the goal to localize and classify individual objects.

The classification result serves as the basis for the sorting decision. Fast-switching pneumatic valves are lined up in an array orthogonal to the main transport direction of the conveyor belt. In this array, individual nozzles can be activated to eject individual particles. Assuming a constant speed of the bulk material due to the transport with the conveyor belt, a fixed delay time between detection and ejection can be set. The number of activated nozzles as well as the duration of activation depends on the detected object size.

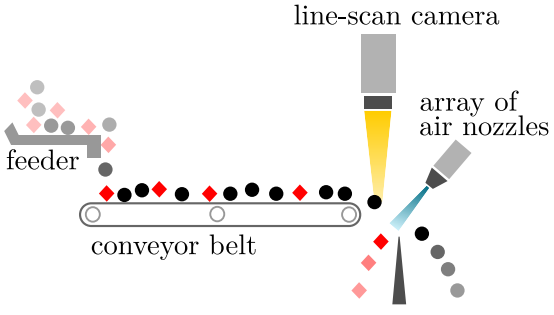

Figure 1: Illustration of a sensor-based sorting system consisting of a conveyor belt and a line-scan camera.

The sorting quality achieved depends primarily on the classification accuracy of the bulk material objects and the correct physical ejection. Especially, the timing and the accuracy of the separation can be influenced and optimized by software with the help of several parameters depending on material and setup.

3.2 Anomaly detection using autoencoder

Anomaly detection [17, 18] based on machine learning methods means to obtain a model that captures the normal behavior trained from positive i.e. anomaly-free data and testing it on test data (data that is not covered by the training dataset) to check if the trained model fits the test data or not. If the test data is inconsistent with the trained model, this means that an anomaly is detected. One popular approach is utilizing dimensionality reduction techniques for anomaly detection. The basic assumption is that the input variables are correlated with each other and the machine learning model can be used to embed this data into a lower dimensional space, where normal and abnormal data appears to be significantly different [18]. There are several methods for dimensionality reduction available, whereas the most known are PCA and kernel PCA [19] and autoencoders [20] as recent dimensionality reduction technique.

Basically, autoencoders are artificial neural networks [21], whose output is aimed to be a reconstruction of its input. The architecture is set as such, that between the input layer and the output layer, there is a bottleneck or representation layer with less neurons compared to its input size [20]. An autoencoder can be split into an encoder and a decoder part. In general both the encoder and the decoder can be build with multiple hidden layers. Within the encoder, the input vectors $x_i \in \mathbb{R}^d$ are reduced to dimension m with $m \ll d$ neurons to build the representation. Within the hidden layers, the activation $a_i \in \mathbb{R}$ of the neuron i is defined in [21] as

$$a_i = \sigma\left(\sum_{j=1}^{n} w_j a_j + b\right) \tag{1}$$

with $w \in \mathbb{R}^n$ weighing the inputs of the neuron, $b \in \mathbb{R}$ the bias, and σ the (non-)linear activation function. Since $m \ll d$, the input vector is encoded to a lower-dimensional space. The decoder takes the representation layer and reconstructs the original space resulting in the output vector $x_i' \in \mathbb{R}^d$. The hidden layers in the decoder are defined in the same way as for the encoder. As noted before, the aim of an autoencoder

is to minimize the difference between original input and its reconstruction. This difference is called reconstruction error $\epsilon \in \mathbb{R}$, which is computed using the euclidean distance [22]:

$$\epsilon = \sqrt{(x_i - x_i')^T \cdot (x_i - x_i')}. \quad (2)$$

The inputs are then labelled based on the reconstruction error: if ϵ is greater than a threshold value, then the input is labelled as "Anomaly", otherwise as "Normal". The threshold value can either be defined by an expert or derived by taking the average of all reconstruction errors.

3.3 Explainable artificial intelligence methods

As mentioned before, machine learning (including deep learning) based systems deliver excellent results in many areas [23]. However, the complex models such as DNNs, SVMs, etc., are considered as black-boxes, as their internal workings are difficult to understand. While delivering outstanding results, these models do not provide explanations (or an explanation) for their predictions. In order to strengthen a potential user's trust in the models, XAI methods can be used.

XAI methods can be distinguished on the basis of various criteria [24]:

Model-specific interpretability: The XAI method is only compatible with certain prediction models. That is, these techniques each apply to a single type of machine learning model because they depend on the internal structure of that model. The benefits of using model-specific methods include enabling us to gain a deeper understanding of the decision by revealing the internal workings of the model on the one hand and the ability to build more specialized explainable models on the other hand. However, the drawback of such models is that the performance of the model is compromised since the model's structure must be entirely overhauled in order to reconstruct or retrain the model itself. Model-specific methods are thus preferred in use cases where the model's performance is not crucial.

Model-agnostic interpretability: The explanatory component and the machine learning models are independent from each other. That is, they do not take into account the internal structure of the model. As Ribeiro et al. have described in [25], this approach offers a great advantage: flexibility. This means that these techniques can be applied to any model. Further, these methods will not affect the performance of the model as they do not require retraining the model. Model-agnostic methods are thus preferred on high-performing, but complex models which are often referred to as black-box models.

The focus of this paper is on the development and investigation of model-agnostic methods, as these allow for faster adaption to new machine learning models. XAI methods can also be classified based on their scope into local and global explanations as shown in Figure 2 [24]:

Local explanations describe a single predictive outcome over the entire model. Thus, the conditional interactions between dependent and independent variables are explained in terms of a single prediction that takes into account the surrounding feature space. In essence, they illustrate how the outcomes of the model (e.g., predictions) change when the values of specific features change within the surrounding feature space.

Global explanations describe the behaviour of the entire model, such as the weight of the features, i.e., the conditional interactions between dependent and independent variables are explained over the entire dataset. In essence, they provide a complete view on the working of the model, e.g., through listing the features that determine the outcomes of the model.

Often used methods for explaining neural networks are either saliency or feature attribution methods [24]:

Saliency methods: These methods are well suited to visualise important regions within the network and also weights that were crucial for the model to make a prediction. The explanation, however, varies every time the process is carried out, even for the same instance, which makes it unreliable [26].

Feature attribution methods: These methods work directly on a subset of the entire dataset in such a way to find out the explanatory power (relevance importance) of each input variable with respect to the target variable.

Due to the unreliability of saliency methods, our focus in this study is on feature attribution methods for

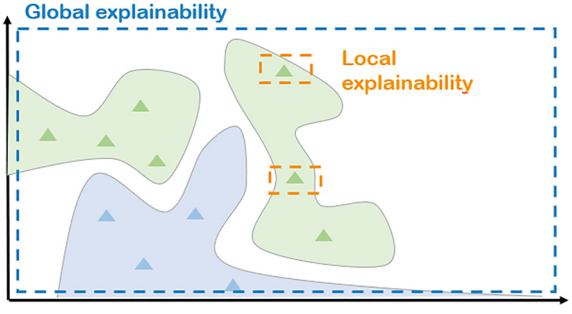

Figure 2: Global and local explainability.

generating explanations. In the following, the adapted XAI explainers for our application are described.

3.3.1 Shapley additive explanations (SHAP)

SHAP [27] is a model-agnostic local approach for generating explanations. It can be used for explaining the prediction of any model by computing the contribution of each feature to the prediction. Essentially, it is derived from the concepts of cooperative game theory. The field of cooperative game theory explains and reasons about the fair distribute of rewards among cooperating players.

One of the main components of the SHAP method is a game theoretic concept called "Shapley value", developed by Shapley [28] in order to be able to assess the influence of a player to form coalitions. According to game theory, any "strategy" combination that all "players" agree upon is accompanied by some sort of "reward". To apply this method to the explainability of machine learning methods, the following analogies are made: the results of the procedures correspond to the "strategies", the "players" are the features, and the "reward" is the quality of the result. Similar to how the Shapley value identifies a player's contribution to the game, SHAP identifies a feature's impact on the overall prediction.

The SHAP value is then defined as the weighted average of the marginal contributions over all possible $|F|!$ coalitions [24] according to

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f(x_{S \cup \{i\}}) - f(x_S)], \quad (3)$$

where ϕ_i is the Shapley value for feature i, F the set of all features, S a subset of F, $f(x_{S\cup\{i\}})$ is the model prediction with feature i and $f(x_S)$ the model prediction without feature i.

It can be challenging to calculate SHAP values precisely, because for a set of f features, 2^f subsets should be analyzed for each feature in order to compute the SHAP values. This results in exponentially longer run-times, when more features are added. Therefore, SHAP approximates the original model $f(x)$ and generates an explanation model which is a linear function of binary variables z'_i:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i = \text{Bias} + \sum \text{features contribution}, \quad (4)$$

where $g(z')$ is a local surrogate model and ϕ_i illustrates the contribution of each feature i to the prediction.

Depending on the specific use case, SHAP offers various approximations. For our application, we employed two: KernelSHAP and DeepSHAP. KernelSHAP is a combination of Linear LIME [25] and SHAP, whereas DeepSHAP is a combination of the DeepLift [29] and SHAP algorithms. Linear LIME and DeepLift are used to approximate the SHAP values in order to linearise the non-linear elements of a neural network.

3.3.2 Local interpretable model-agnostic explanations (LIME)

In contrast to the SHAP method, LIME [25] explains individual predictions of black-box models, by approximating them locally with an interpretable surrogate model. The main idea behind this method is that it tests how a prediction changes when variation in the data is provided to the model. Additionally, it generates a new dataset with modified samples and corresponding predictions. Then on this dataset, LIME builds a model that is weighted according to how close the sampled instances are to the instance of interest. This process gives an idea of which feature has more influence on the model's prediction. LIME assumes, that a simpler more transparent model can be used to explain the prediction of a complex model locally.

In order to approximate the original model f, LIME minimises the following objective function

$$\xi(x) = \underset{g \in G}{\arg\min} L(f, g, \pi_x) + \Omega(g), \quad (5)$$

where g is an interpretable model from a set of Models G, x is the original data instance, and π_x is the proximity measure from all permutations to the original observation. The $L(f, g, \pi_x)$ term represents the measure of unfaithfulness of g in approximating f in the locality defined by π. The $\Omega(g)$ term is a measurement of the model's complexity regarding an interpretable model g.

3.4 Applying XAI to an autoencoder

Since an autoencoder is an unsupervised machine learning model, as was explained in Section 3.2, feature attribution XAI techniques mentioned in Section 3.3 cannot be utilized directly to explain the predictions of such a black-box model. The feature attribution XAI techniques such as SHAP or LIME always anticipate a black-box model to predict an output in the form of prediction probabilities (in the case of classification problem) or a number (in the case of regression problem). The prediction of an autoencoder, however, is the best possible reconstruction of the input. This makes the integration of XAI into an autoecoder challenging. Therefore

we adapted the methodology proposed in [15] for our application to explain anomalies detected by an autoencoder in the sorting process.

The pipeline to generate XAI explanations on an unsupervised machine learning model is shown in Figure 3. The dataset $\mathcal{D}$ of size N, described by

$$\mathcal{D} = \{(x_i) \mid x_i \in \mathcal{X}\}_{i=1}^{N} \tag{6}$$

consists of the feature vectors $x_i \in \mathcal{X}$ with $\mathcal{X}$ being the possible inputs. It is passed on to train an unsupervised machine learning model, in our case an autoencoder. The unsupervised model is denoted by M_θ, where $\theta \in \Theta$ are its trained parameters. It is used to generate target labels $y_i \in \mathcal{Y}$ for the dataset by comparing the reconstruction error ϵ with a threshold. That means, if ϵ is greater than a threshold value, then the input is labelled as "Anomaly", otherwise as "Normal", as mentioned in Section 3.2. With this, we obtain a labelled dataset $\mathcal{D}'$ of size N

$$\mathcal{D}' = \{(x_i, y_i)\}_{i=1}^{N} . \tag{7}$$

The input x_i along with $M_\theta(x_i)$ is then fed to the surrogate model, which is a supervised machine learning model because of the presence of target labels. Here, the surrogate model is again a *Neural Network* (*NN*), denoted by N_ρ, given by its parameters ρ. The objective of this surrogate model is to minimize the loss function $L(\rho, D')$. This surrogate model is then used to generate explanations using feature attribution XAI techniques described in Section 3.3. It is assumed that the surrogate model gives high relevance to the same features as the unsupervised model—autoencoder—when approximating the decision behavior. The key benefit of this approach is its dual capability of generating both local and global explanations, which is an advantage not offered by the approach proposed in [14]. Furthermore, the pipeline illustrated in Figure 3 may be applied to any unsupervised model, such as K-Means clustering, principal component analysis (PCA), etc., making it a model-agnostic solution.

4 Examined sorting system

At first, the experimental setup of sensor-based sorting system, autoencoder, and XAI methods are described in this section. Then we discuss the initial results of XAI methods.

4.1 Experimental sensor-based sorting system

The sensor-based sorter used in the following investigation is a laboratory-scale, modular version of a full sized industrial sorting system. A thorough description is provided in [30]. The setup is equipped with a line-scan color camera. Additionally, it includes a vibratory feeder which distributes the bulk material on the conveyor belt. The conveyor belt is set to run at a constant velocity of approximately 1 ms^{-1}. For separation, a nozzle array consisting of 32 individual valves is mounted behind the conveyor belt. Each nozzle can be activated individually and operates at a gauge pressure of 1.2–1.6 bar. The setup allows the separation of a material stream into two different material streams, realizing a binary sorting task.

During operation, a variety of different statistics about the sorting process are collected, stored or output live via available interfaces. The period of time during which data is accumulated is called the "StatisticInterval" and is initially set to 5 s. One statistic, called "ValveBarActivation", counts how many times the nozzles have been activated during the set statistic interval. The result is a 32 dimensional array, generated every 5 s.

For the sorting experiments, red brick particles are separated from white sand-lime brick particles. The objects have a size of 3–5 mm. The vibratory feeder is adjusted so that an occupancy density on the conveyor belt of approx. 40% resulted. Three different scenarios are covered:

1. Normal condition: all objects are approximately equally distributed on the conveyor belt.
2. Uneven distribution left and right: Due to incorrect feeding, a skewed normal distribution of objects on the conveyor occurs.
3. Throughput error: occupancy changes as more objects fall on the conveyor belt.

An autoencoder is used to model the normal behaviour (following the first scenario: normal condition) of the nozzle activations by using the "ValveBarActivation" as an input.

4.2 Experiments and initial results

The experimental setup and initial results are presented in this section. First, the setup used to carry out our experiments is outlined, which includes the architecture of an unsupervised NN as well as a surrogate NN. Thereafter, the initial results of an autoencoder and its corresponding XAI explanations are presented.

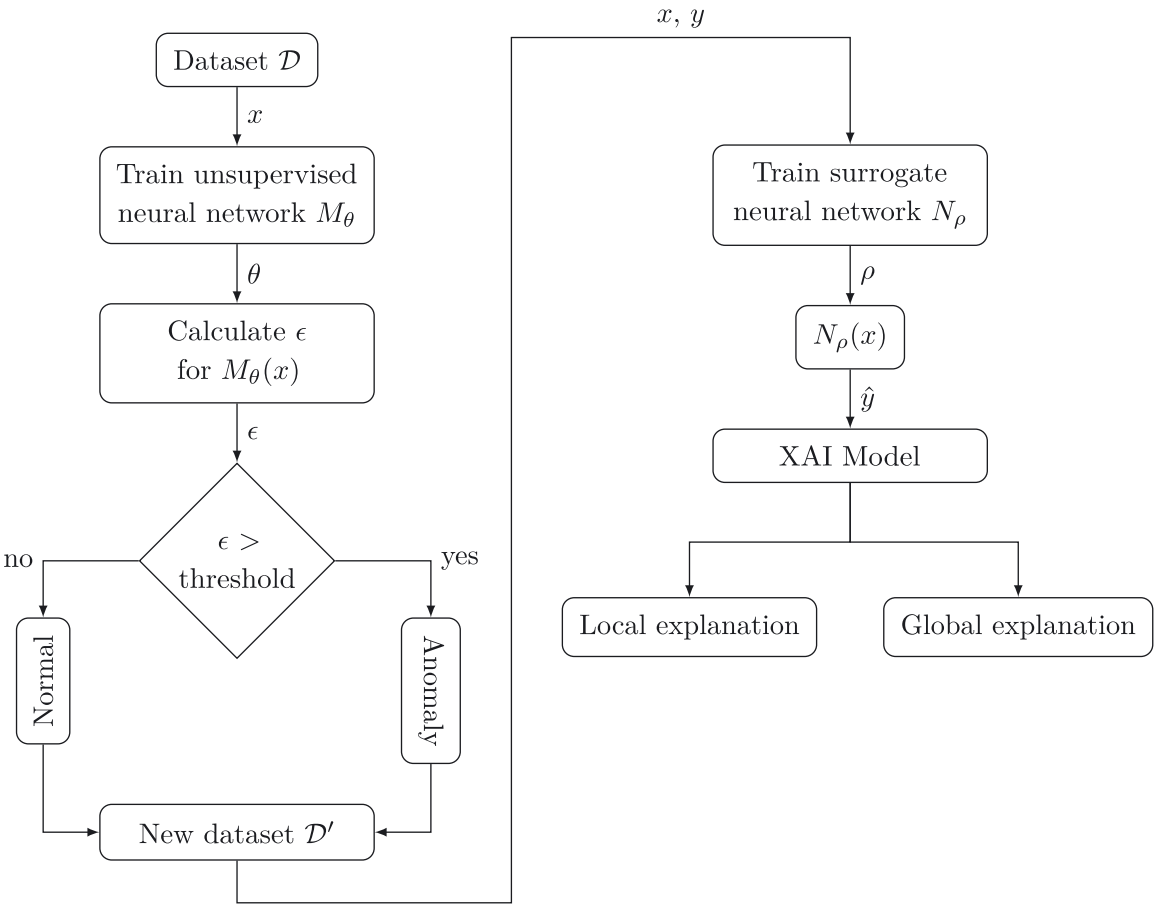

Figure 3: Pipeline to generate XAI explanations on an unsupervised model.

4.2.1 Experimental setup

The real-world data was collected on the demonstrator described above. The dataset consists of 605 samples, where each sample has 32 features. These features basically represent the number of times the nozzles were activated under the supervision of respective evaluation unit during the sorting process (the "ValveBarActivation"). The goal here is to detect anomalies specifically in the material feeding during the sorting process using AI. For that purpose, the aforementioned autoencoder is used. The network of this model consists of layers such as normalizer, input, and multiple dense layers. Keras "*Normalization*" method is used as a normalizer to make sure that the input features are on a similar scale, which stabilizes the gradient descent step. *Mean squared error* is used as a loss function and *tanh* as an activation function in the three hidden layers. The representation layer has 128 neurons, while the other two hidden layers (one for the encoder and one for the decoder part) have 256 neurons. The autoencoder model uses the Adam optimization algorithm with a learning-rate of 0.001. The available data is split into 85% for training and 15% for testing.

The surrogate model is then trained with a training set together with the target label derived from the reconstruction error of the autoencoder model. The network of the surrogate model consists of the following layers: a normalizer (similar to the one in an autoencoder), three dense and dropout combinations, and an output layer. The input to the network depends on the number of features present in the dataset, which is 32 in our case, the first dense layer has 16, the second 8 and the last 4 neurons. A *sparse categorical cross-entropy* is used as a classification loss function, *tanh* as an activation function in the hidden layer and *softmax* for the output. The surrogate model uses the Adam optimization algorithm with a learning-rate of 0.001. The aim is to approximate the decision behavior of an autoencoder by classifying whether an input is an anomaly or normal. The fidelity of the surrogate model also depends on the threshold used for generating the labels. For the following analysis, the fidelity is 0.86. This trained surrogate model is then used to explain the predictions of an autoencoder

using XAI techniques. To explain the output of the surrogate model, the number of background samples for KernelSHAP, DeepSHAP, and LIME was chosen as ∼257 (50% of training set).

4.2.2 Explanation results

In order to analyze the local explanations of XAI methods, let us consider a single instance or an observation from the dataset as shown in Figure 4. The instance is of the form (1, 32), which contains one sample with 32 features. The features (directly corresponding to the nozzles) from 0 to 11 are represented by numbers greater than zero, indicating that they were activated and the other input features are set to zero, indicating that they were not activated, during the sorting process. The autoencoder is used for anomaly detection and its result for this instance is "Anomaly", and the prediction of a surrogate model is an "Anomaly" as well with a prediction probability of 0.88. On a selected observation, each of the XAI explainers produces a single plot as a local explanation: a force plot in the case of KernelSHAP and DeepShap, and a feature importance plot in the case of LIME. These plots illustrate which features had the most influence on the model's prediction for a single observation.

The output of the KernelSHAP explainer is shown in Figure 5a. We can observe that the predicted value for "Anomaly" is 0.88 and the corresponding base value is 0.24. The base value is "the average model output over the training dataset that we passed". The fractional values are the amount of importance contributing towards the prediction "Anomaly" by pushing the model output higher (red) or lower (blue) from the base value. Here, red indicates positive contribution towards the decision and blue indicates negative contribution. The biggest impact comes from the *Feature* 0, where its visual size indicates the magnitude of the feature's effect.

Similar local explanations are obtained from the DeepSHAP and LIME explainers as shown in Figure 5b and c respectively. According to the DeepSHAP explainer, the major impact comes from *Feature* 0 and *Feature* 5, whereas the LIME explainer indicated that the features such as *Feature* 0 and *Feature* 4 are relatively dominant.

Further, in order to analyze the global explanations of XAI explainers, we examine the entire training dataset described in Section 4.2.1. For this analysis, KernelSHAP and DeepSHAP are used for generating global explanations. The output of both explainers (top 10 features) towards the prediction class "Anomaly" are shown in Figure 6. It can be seen that the outputs of both explainers are similar, indicating *Feature* 0 has the most influence on the model predictions followed by *Feature* 29 and *Feature* 30 for KernelSHAP and vice-versa for DeepSHAP. The follow-up features consist of a combination of *Feature* 1, *Feature* 5, *Feature* 10 and *Feature* 16.

However, it is to be noted that the feature importance plot contains no information beyond the importances. It does not, for instance, indicate the relationship between the value of a feature and the impact on the model's prediction. For this purpose, a more informative plot such as a summary plot is produced as shown in Figure 6c and d for KernelSHAP and DeepSHAP respectively. Each point on the summary plot is Shapley value for a feature and an instance. The Shapley values are shown on the x-axis and features along with their values are shown on the y-axis. The color scale represents feature values from low (blue) to high (red). The features are listed in descending order of importance.

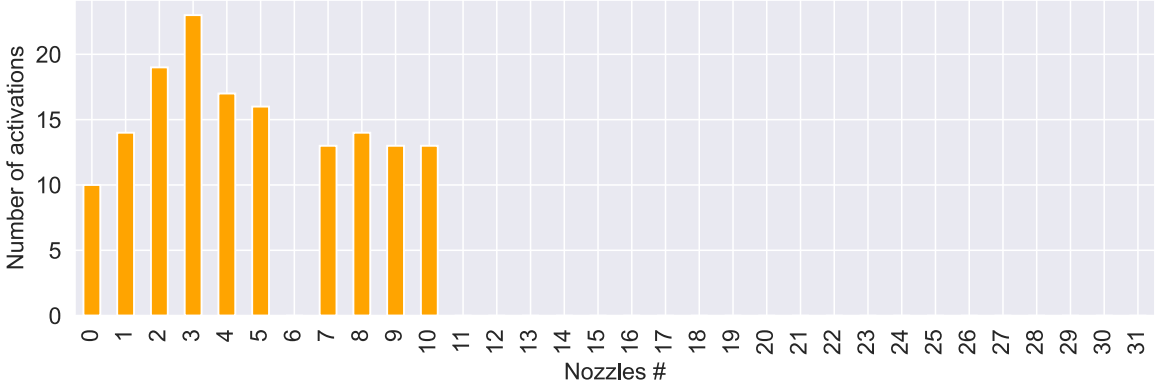

Figure 4: Representation of a selected instance from the dataset.

Figure 5: XAI local explanations for the prediction "Anomaly". Here, features represent the nozzles that are activated and deactivated by the respective sensor during the sorting process. (a) KernelSHAP explanation in the form of a force plot. (b) DeepSHAP explanation in the form of a force plot. (c) LIME explanation in the form of a feature importance plot.

As can be observed from the summary plots of both the explainers, features with higher value – such as *Feature* 0, *Feature* 1, *Feature* 29, and *Feature* 30 – increase the prediction towards anomaly whereas features with lower values decrease it. This provides the first indications as to how the value of a feature and its impact on the prediction relate to one another. But to see the exact form of the relationship and the dependencies between the features, we employ SHAP dependence plots.

The dependency plot for *Feature* 0 and *Feature* 1 interactions is shown in Figure 7. It can be seen that higher values of *Feature* 0 with higher values of *Feature* 1 are more likely to contribute towards the prediction class "Anomaly" than lower values. One possible reason for that is due to throughput error. More about this is discussed in the Section 4.2.3. Similarly, the interactions between other interesting features can be obtained using SHAP dependence plots.

4.2.3 Discussion

According to the local and global explanations of XAI explainers, the features ranging from 0 to 5 (nozzles in the left corner), *Feature* 30 and *Feature* 29 (nozzles in the right corner) have more influence on the prediction class "Anomaly" than the rest of the features. This could be due to the following reasons, as per our knowledge on sorting mechanism and the analyzed scenarios:

– The conveyor belt, that carries the sorting materials before sorting, may be inclined to one side, or the feeder is not working correctly (which is why, for example,

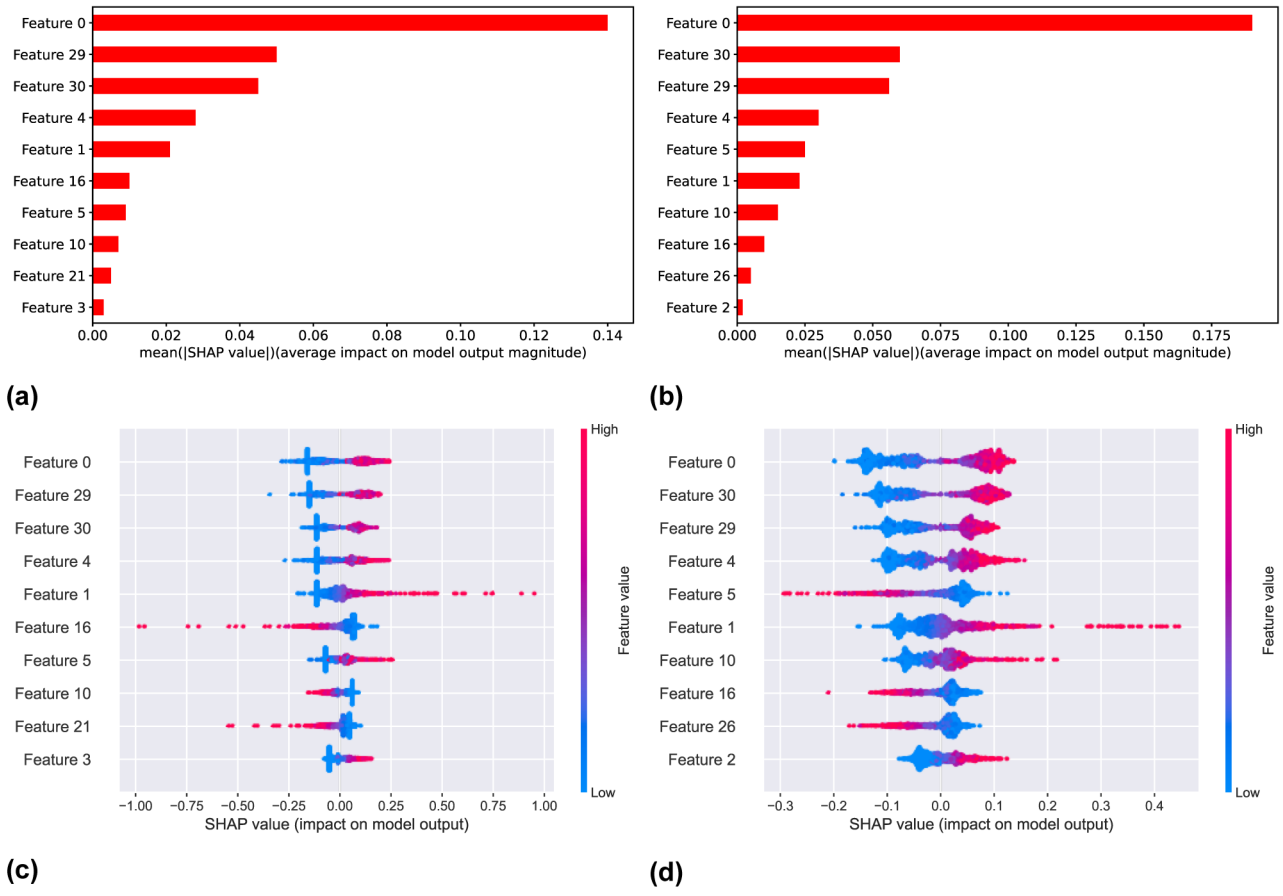

Figure 6: XAI global explanations for "Anomaly" predictions. Here, features represent the nozzles that are activated and deactivated by the respective sensor during the sorting process. (a) KernelSHAP feature importance plot. (b) DeepSHAP feature importance plot. (c) KernelSHAP summary plot. (d) DeepSHAP summary plot.

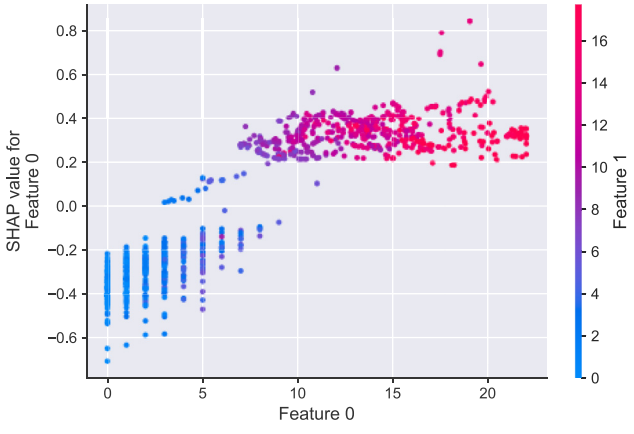

Figure 7: SHAP dependency plot for *Feature* 0 and *Feature* 1 interactions.

the selected instance (Figure 4) is considered as an "Anomaly", where only the nozzles from 0 to 11 are activated) and the materials are not equally distributed.

– This behavior could also indicate problems with the sorting mechanism. For example, if some of the nozzles fails to activate during the sorting process, then that would cause abnormal behavior and would worsen the sorting quality.

On the other hand, the global summary plots as well as dependence plots indicate that features with higher values would push the model's prediction towards an anomaly. That means, if the nozzles are activated more often than anticipated during the designated statistic interval, then that would lead to an anomalous situation (throughput error). One possible scenario could be the following: If the throughput increases, more objects fall on to the conveyor belt. In that case, the objects overlap on one another making it difficult for sorting. Nozzles activate more frequently as a result, which lowers the sorting quality.

It is believed that these kind of local and global explanations from XAI explainers support the operator to understand the behavior of the sorting mechanism and, on the

other hand, help the analysts to improve their anomaly detection models for sorting applications. Further, with the use of XAI explanations, the operator can also detect these anomalies or abnormal behaviors in advance and prevent sorting machines from getting damaged.

5 Conclusions and future work

In this work, we gave a brief introduction on sensor-based sorting systems, the used demonstrator with an autoencoder as foundation for anomaly detection, and XAI methods and their application to the given sorting problem. Three XAI Methods (KernelSHAP, DeepSHAP, and LIME) have been adapted and implemented for explaining the anomalies detected by an autoencoder. The results illustrate the influence of each feature (nozzles in this use case) on the model's decision. This helps the operator to understand the behavior of the sorting mechanism and detect the anomalies in advance to prevent damages and defects. Further, the analysts could improve the performance of their anomaly detection models. As a result, XAI explanations give insights into the autoencoder's reasonings by making the whole system more transparent.

However, it is to be noted that while XAI methods offer a level of transparency and interpretability, the explanations provided by these models may not always be accurate or complete. Different XAI methods can provide different explanations for the same instance, which can make it challenging for human experts to select the appropriate explanation. Therefore, it's important to evaluate the explanations provided by XAI methods using both computational methods [4] and human-subject studies [9] to ensure their accuracy and completeness. Future work in this field will aim to improve the evaluation of XAI explanations to ensure their usefulness and credibility.

Author contributions: All the authors have accepted responsibility for the entire content of this submitted manuscript and approved submission.
Research funding: Funded by the Fraunhofer Research Center for Machine Learning within the Fraunhofer Cluster of Excellence Cognitive Internet Technologies.
Conflict of interest statement: The authors declare no conflicts of interest regarding this article.

References

[1] S. J. Russell and N. Peter, *Artificial Intelligence: A Modern Approach*, 3rd ed. Boston, Prentice-Hall Series in Artificial Intelligence. Pearson, 2010.

[2] V. Buhrmester, D. Münch, and M. Arens, "Analysis of explainers of black box deep neural networks for computer vision: a survey," CoRR, abs/1911.12116, 2019, Available at: http://arxiv.org/abs/1911.12116.

[3] Y. H. Sheu, "Illuminating the black box: interpreting deep neural network models for psychiatric research," *Front. Psychiatry*, vol. 11, p. 1091, 2020.

[4] M. Veerappa, M. Anneken, N. Burkart, and M. Huber, "Validation of XAI explanations for multivariate time series classification in the maritime domain," *J. Comput. Sci.*, vol. 58, p. 101539, 2021.

[5] H. Wilts, B. R. Garcia, R. G. Garlito, L. S. Gómez, and E. G. Prieto, "Artificial intelligence in the sorting of municipal waste as an enabler of the circular economy," *Resources*, vol. 10, no. 4, p. 28, 2021.

[6] I. H. Sarker, "Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions," *SN Comput. Sci.*, vol. 2, p. 420, 2021.

[7] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable ai: A review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, 2021, https://doi.org/10.3390/e23010018.

[8] U. Schlegel, H. Arnout, M. El-Assady, D. Oelke, and D. A. Keim, "Towards a rigorous evaluation of XAI methods on time series," in *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 4197—4201.

[9] N. Burkart, M. Huber, and M. Anneken, "Supported decision-making by explainable predictions of ship trajectories," in *15th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2020)*, 2021, pp. 44—54.

[10] M. Veerappa, M. Anneken, and N. Burkart, "Evaluation of interpretable association rule mining methods on time-series in the maritime domain," in *Pattern Recognition. ICPR International Workshops and Challenges*, A. Del Bimbo, R. Cucchiara, S. Sclaroff, G. M. Farinella, T. Mei, M. Bertini, H. J. Escalante, and R. Vezzani, Eds., Cham, Springer International Publishing, 2021, pp. 204—218.

[11] N. Burkart and M. F. Huber, "A survey on the explainability of supervised machine learning," *J. Artif. Intell. Res.*, vol. 70, pp. 245—317, 2021.

[12] G. Sofianidis, J. M. Rožanec, D. Mladenić, and D. Kyriazis, *A Review of Explainable Artificial Intelligence In Manufacturing*, 2021, Available at: https://arxiv.org/abs/2107.02295.

[13] V. Nasteski, "An overview of the supervised machine learning methods," *Horiz. B*, vol. 4, pp. 51—62, 2017.

[14] L. F. Antwarg, R. Miller, B. Shapira, and L. Rokach, "Explaining anomalies detected by autoencoders using shapley additive explanations," *Expert Syst. Appl.*, vol. 186, p. 115736, 2021.

[15] M. Schultz, N. Gnoss, and M. Tropmann-Frick, *XAI in the Audit Domain — Explaining an Autoencoder Model for Anomaly Detection*, Nuremberg, Germany, Wirtschaftsinformatik 2022 Proceedings. 1., 2022.

[16] K. Roshan and A. Zafar, "Utilizing XAI technique to improve autoencoder based model for computer network anomaly detection with shapley additive explanation (SHAP)," *Int. J. Comput. Netw. Commun.*, vol. 13, no. 6, pp. 109—128, 2021.

[17] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," *ACM Comput. Surv.*, vol. 41, no. 15, pp. 1—58, 2009.
[18] S. Omar, A. Ngadi, and H. H. Jebur, "Machine learning techniques for anomaly detection: an overview," *Int. J. Comput. Appl.*, vol. 79, no. 2, pp. 33—41, 2013.
[19] L. Chapel and C. Friguet, "Anomaly detection with score functions based on the reconstruction error of the kernel PCA," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2014, pp. 227—241.
[20] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," in *2018 Wireless Telecommunications Symposium (WTS)*, 2018, pp. 1—5.
[21] C. C. Aggarwal, *Neural Networks and Deep Learning — A Textbook*, New York, Springer, 2018.
[22] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli, "Euclidean distance matrices: essential theory, algorithms, and applications," *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 12—30, 2015.
[23] L. Alzubaidi, J. Zhang, A. Humaidi, et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, p. 53, 2021.
[24] C. Molnar, *Interpretable Machine Learning*, 2019.
[25] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135—1144.
[26] P. J. Kindermans, S. Hooker, J. Adebayo, et al., "The (Un)reliability of Saliency methods," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, 1st ed. Switzerland, Springer Publishing Company, Incorporated, 2017, pp. 267—280.
[27] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," *Adv. Neural. Inf. Process. Syst.*, vol. 30, pp. 4765—4774, 2017.
[28] L. S. Shapley, "A value for n-person games," *Contrib. Theory Game*, vol. 2, no. 28, pp. 307—317, 1953.
[29] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International Conference on Machine Learning*, 2017, pp. 3145—3153.
[30] G. Maier, F. Pfaff, C. Pieper, et al., "Experimental evaluation of a novel sensor-based sorting approach featuring predictive real-time multiobject tracking," *IEEE Trans. Ind. Electron.*, vol. 68, no. 2, pp. 1548—1559, 2021.

Bionotes

Mathias Anneken received a master's degree in Electrical Engineering and Information Technology after studying at the Karlsruhe Institute of Technology (KIT). He joined Fraunhofer IOSB in 2014 and is currently finishing his PhD in cooperation with the KIT in the fields of artificial intelligence, machine learning, and anomaly detection in the maritime domain. Since July 2022 Mathias Anneken leads the research group Applied Explainable Artificial Intelligence at Fraunhofer IOSB.

Manjunatha Veerappa is a research employee at Fraunhofer IOSB where he is responsible for developing XAI algorithms. Before joining Fraunhofer, Manjunatha Veerappa received a master's degree specializing in real-time data processing and pattern recognition. Prior to this, he worked at Cognizant Technology Solutions as a programmer analyst with a primary focus on interpreting the datasets using statistical tools.

Marco F. Huber received his diploma, Ph.D., and habilitation degrees in computer science from the Karlsruhe Institute of Technology (KIT), Germany, in 2006, 2009, and 2015, respectively. From June 2009 to May 2011, he was leading the research group "Variable Image Acquisition and Processing" of the Fraunhofer IOSB, Karlsruhe, Germany. Subsequently, he was Senior Researcher with AGT International, Darmstadt, Germany, until March 2015. From April 2015 to September 2018, he was responsible for product development and data science services of the Katana division at USU Software AG, Karlsruhe, Germany. At the same time he was adjunct professor of computer science with the KIT. Since October 2018 he is full professor with the University of Stuttgart. He further is director of the Center for Cyber Cognitive Intelligence (CCI) and of the Department for Image and Signal Processing with Fraunhofer IPA in Stuttgart, Germany. His research interests include machine learning, planning and decision making, image processing, data analytics, and robotics. He has authored or co-authored more than 100 publications in various high-ranking journals, books, and conferences, and holds two U.S. patents and one EU patent.

Christian Kühnert received is diploma in electrical engineering in 2008 from TU Darmstadt. In the same year he joined Fraunhofer IOSB. In 2013 Christian Kühnert finished his PhD at the Karlsruhe Institute of Technology (KIT). Since 2019 he is senior scientist at Fraunhofer IOSB.

Felix Kronenwett studied at the DHBW Karlsruhe and continued his studies in Electrical Engineering and Information Technology at the Karlsruhe Institute of Technology (KIT). He joined Fraunhofer IOSB in 2021.

Georg Maier received the M.Sc. degree in computer science from Utrecht University, Utrecht, The Netherlands, in 2013. Afterwards, he joined the Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, Karlsruhe, Germany. In 2021 he finished is PhD at the Karlsruhe Institute of Technology (KIT). His research interests include different aspects of image processing, in particular algorithmic aspects, with a focus on real-time capabilities.